



# Logiciel de Génération et Représentation de Structures Cristallines

Alain Soyer, Alexandre Rimsky

## ► To cite this version:

Alain Soyer, Alexandre Rimsky. Logiciel de Génération et Représentation de Structures Cristallines. Journal of Applied Crystallography, 1992, 25 (2), pp.214-220. 10.1107/S0021889891012086 . hal-01214809

**HAL Id: hal-01214809**

**<https://hal.science/hal-01214809>**

Submitted on 23 Nov 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## Logiciel de Génération et Représentation de Structures Cristallines

PAR ALAIN SOYER ET ALEXANDRE RIMSKY

Laboratoire de Minéralogie-Cristallographie associé au CNRS, UA09, Université P. et M. Curie,  
Tour 16, 2ème étage, 4 place Jussieu, 75252 Paris CEDEX 05, France

(Reçu le 18 juin 1991; accepté le 21 novembre 1991)

### Abstract

Software has been written in Fortran77 for the space-filling representation of crystal structures. It generates atomic positions starting from the asymmetric unit and taking into account the 230 possible space groups. It then computes realistic shaded images in correct perspective with the hidden part removed. The main differences from existing programs are the ability to represent thermal ellipsoids and good portability to commonly used workstations. Details are given about the organization and operation of the program and some applications.

### 1. Introduction

Il existe déjà de nombreux programmes de représentation de structures cristallines ou moléculaires dont beaucoup ont été cités dans les publications de Max (1983) and Keller (1989). En comparant les caractéristiques de ces logiciels nous avons constaté, à côté des possibilités et avantages propres à chacun d'eux, les points faibles suivants:

(a) Environ 50% des programmes ne prennent pas en compte la perspective, se contentant de projections orthogonales. Ceci est gênant, en particulier si on veut faire des paires stéréographiques de qualité.

(b) Un pourcentage encore plus grand n'est pas (ou difficilement) portable. Ceci vient du fait que les traceurs et les écrans des différents constructeurs ont des langages graphiques divers. De plus pour des raisons de performances des auteurs ont utilisé des langages de bas niveau comme l'assembleur.

(c) Seul le programme *ORTEP* de Johnson (1965, 1969) permettait jusqu'à présent de dessiner les ellipsoïdes d'agitation thermique.

(d) Beaucoup de programmes sont conçus pour des représentations de molécules isolées, plus rarement pour des structures cristallographiques.

Dans notre laboratoire, un programme de A. Lesk (provenant lui-même d'un programme de D. Richardson) a d'abord été adapté et utilisé. F. Morris a ensuite écrit *QUADR*, un logiciel général de représentation d'objets formés de quadriques (voir Epelboin, Guidi-Morosini, Morris, Rimsky & Soyer, 1986). Puis nous avons développé notre propre logiciel en langage

Fortran77 standard, valide pour les 230 groupes d'espace cristallographiques et permettant la représentation des ellipsoïdes d'agitation thermique sous forme d'images réalistes avec perspective et ombrage. Cette réalisation a été faite pour des petites stations de travail, qui sont devenues guère plus chères que de gros micro-ordinateurs (sans en avoir les limitations comme, par exemple, de mémoire) et en privilégiant la portabilité. Ce logiciel peut être utile aux chercheurs pour générer des structures cristallines afin d'effectuer des calculs ou de les représenter sur écran graphique à balayage, ainsi qu'à des fins pédagogiques pour montrer aux étudiants en cristallographie comment construire un cristal à partir de l'unité asymétrique en appliquant les éléments générateurs du groupe d'espace.

### 2. Organisation générale

L'ensemble des trois programmes permettant de générer et représenter le contenu d'un cristal est schématisé sur la Fig. 1.

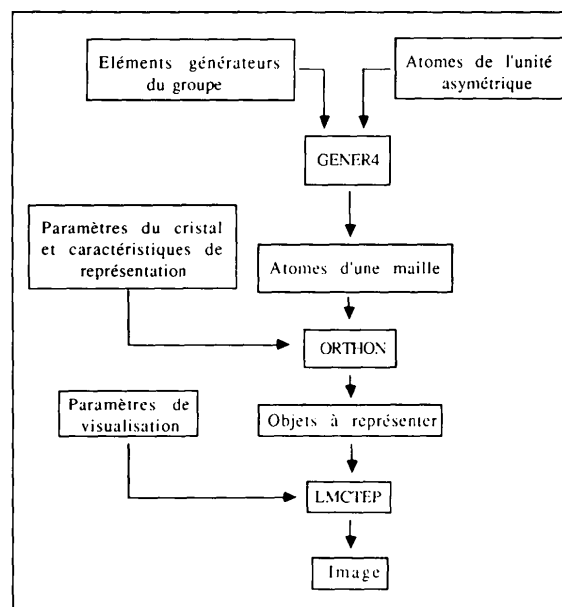


Fig. 1. Organisation générale du logiciel.

*GENER4* est un programme de génération de positions atomiques symétriquement équivalentes, avec élimination automatique des positions dégénérées. Il est valide pour les 230 groupes d'espace cristallographiques. En plus de la génération des positions atomiques équivalentes, grâce à l'utilisation de matrices  $4 \times 4$  représentant les éléments de symétrie du groupe considéré, le programme donne les nouvelles composantes du tenseur d'agitation thermique anisotrope pour chaque atome généré.

Le second programme *ORTHON* permet de construire le contenu de plusieurs mailles en coordonnées orthonormées, et de générer automatiquement les liaisons si on le désire.

Le résultat d'*ORTHON* est visualisé grâce au programme de représentation *LMCTEP*. Il permet la synthèse d'images tridimensionnelles réalistes en perspective vraie, avec possibilité de rotations de la structure représentée et choix de la direction de l'éclairage.

### 3. Le programme de génération *GENER4*

#### 3.1. Calcul des positions équivalentes

D'après les *International Tables for X-ray Crystallography* (1983) une opération de symétrie cristallographique peut être représentée par une matrice  $4 \times 4$  comportant une partie supérieure gauche **W** de dimension  $3 \times 3$  représentant la composante rotation ou symétrie habituelle, et une partie **w** de dimension  $3 \times 1$  représentant la partie translation de l'opération de symétrie.

Pour obtenir le point  $\tilde{\mathbf{X}}$  de coordonnées  $\tilde{x}$ ,  $\tilde{y}$ ,  $\tilde{z}$  équivalent à **X** (position d'un atome de l'unité asymétrique) par application d'une opération de symétrie, on fait subir aux coordonnées du point **X** les transformations suivantes:

(a) Translation (éventuellement nulle) pour amener l'élément de symétrie à l'origine. Cette opération est représentée par la matrice

$$\begin{bmatrix} \mathbf{I} & -\mathbf{p} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -p_1 \\ 0 & 1 & 0 & -p_2 \\ 0 & 0 & 1 & -p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$p_i$  étant les coordonnées du centre de symétrie contenu dans l'élément générateur s'il en a un, sinon un point quelconque de cet élément.

(b) Application de l'élément de symétrie proprement dit

$$\begin{bmatrix} \mathbf{W} & \mathbf{w} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & w_1 \\ W_{21} & W_{22} & W_{23} & w_2 \\ W_{31} & W_{32} & W_{33} & w_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

(c) Translation pour ramener l'élément de symétrie à sa place

$$\begin{bmatrix} \mathbf{I} & \mathbf{p} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_1 \\ 0 & 1 & 0 & p_2 \\ 0 & 0 & 1 & p_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

La séquence des trois transformations se résume à l'application d'une seule matrice **S**, produit des trois matrices précédentes

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{z} \\ 1 \end{bmatrix} = [\mathbf{S}_{ij}] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

#### 3.2. Composantes du tenseur d'agitation thermique

Trueblood (1956) a donné les expressions des composantes du tenseur d'ordre 2 d'agitation thermique après application des opérations de symétrie cristallographiques, sans toutefois en détailler l'obtention. D'autre part, Levy (1956) a explicité les règles de détermination de ces expressions.

Pour notre part il nous a semblé pratique d'utiliser le facteur de température anisotrope sous la forme

$$\exp[-(B_{11}h_1^2 + B_{22}h_2^2 + B_{33}h_3^2 + 2B_{12}h_1h_2 + 2B_{13}h_1h_3 + 2B_{23}h_2h_3)],$$

avec  $\mathbf{h} = h_1\mathbf{a}^* + h_2\mathbf{b}^* + h_3\mathbf{c}^*$  vecteur réflexion de l'espace réciproque.

Cette expression peut en effet aussi s'écrire sous la forme matricielle,

$$[h_1 \ h_2 \ h_3] \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \quad \text{avec } B_{ij} = B_{ji}.$$

Les nouvelles composantes du tenseur d'agitation thermique  $\tilde{\mathbf{B}}$  après application d'une opération de symétrie représentée par la matrice  $3 \times 3$  **W** s'obtiennent alors par les produits matriciels,

$$\tilde{\mathbf{B}} = \mathbf{W}\mathbf{B}\mathbf{W}^T,$$

$\mathbf{W}^T$  étant la transposée de **W**.

#### 3.3. Organisation et fonctionnement de *GENER4*

L'organisation de *GENER4* est résumée sur la Fig. 2. Le programme principal fait d'abord appel au sous-programme *LECGEN* dont le rôle est de lire les données des éléments générateurs. Pour chaque élément de symétrie générateur, *LECGEN* fait appel au sous-programme approprié (à gauche Fig. 2) qui effectue un certain nombre de contrôles sur les données, puis calcule la matrice  $4 \times 4$  associée à ce générateur.

Pour chaque atome de l'unité asymétrique, *GENER4* va effectuer les opérations suivantes:

(a) Lecture des données concernant l'atome.  
 (b) Appel au sous-programme *GENEQU* qui génère les atomes équivalents comme indiqué au §3.1, en appliquant successivement toutes les matrices  $4 \times 4$  associées aux opérations de symétries du groupe. A chaque nouvelle position symétriquement équivalente calculée *GENEQU* fait appel au sous-programme *EXISTE* pour contrôler si ce nouvel atome n'a pas déjà été généré précédemment: l'atome n'est retenu que s'il est nouveau. *GENEQU* fait aussi calculer les nouvelles composantes du tenseur d'agitation thermique par le sous-programme *ROTB*, comme indiqué au §3.2.

(c) Ecriture de tous les atomes générés dans le fichier résultat.

D'après son fonctionnement on voit que le temps d'exécution de *GENER4* sera proportionnel au nombre d'atomes de l'unité asymétrique, ainsi qu'à la somme des ordres de tous les éléments de symétrie générateur. D'autre part la taille mémoire demandée par *GENER4*, qui comporte environ 2400 lignes de Fortran, est de l'ordre de 480 kbytes, ce qui permet de l'exécuter sur n'importe quel ordinateur.

#### 4. Le programme *ORTHON*

Comme indiqué sur la Fig. 1 le fichier résultat de *GENER4* va être repris par le programme *ORTHON* qui permet de générer le contenu de plusieurs mailles en coordonnées orthonormées, avec des atomes sphériques ou des ellipsoïdes, et des liaisons si on le désire. La matrice  $4 \times 4$  associée à un ellipsoïde centré en  $x_0, y_0, z_0$  se calcule de la manière

suivante:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x_0 & -y_0 & -z_0 & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & T_{13} & 0 \\ T_{12} & T_{22} & T_{23} & 0 \\ T_{13} & T_{23} & T_{33} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

où les composantes  $T_{ij}$  dans un repère orthonormé sont déduites des  $B_{ij}$  du repère cristallographique.

Ceci nous donne l'équation générale d'un ellipsoïde:

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{12} & T_{22} & T_{23} & T_{24} \\ T_{13} & T_{23} & T_{33} & T_{34} \\ T_{14} & T_{24} & T_{34} & T_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0,$$

avec:

$$\begin{aligned} T_{14} &= -(T_{11}x_0 + T_{12}y_0 + T_{13}z_0), \\ T_{24} &= -(T_{12}x_0 + T_{22}y_0 + T_{23}z_0); \\ T_{34} &= -(T_{13}x_0 + T_{23}y_0 + T_{33}z_0); \\ T_{44} &= T_{11}x_0^2 + T_{22}y_0^2 + T_{33}z_0^2 + 2(T_{12}x_0y_0 \\ &\quad + T_{13}x_0z_0 + T_{23}y_0z_0) - 1. \end{aligned}$$

Si l'utilisateur a demandé des liaisons, le programme les générera automatiquement sous forme de matrices  $4 \times 4$ , pour les atomes de nature donnée dont la distance est inférieure à la valeur indiquée dans un fichier de données. Pour ce, il part de la matrice associée à un cylindre de rayon  $R$  parallèle à  $Ox$ :

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/R^2 & 0 & 0 \\ 0 & 0 & 1/R^2 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0$$

et il oriente ce cylindre selon la direction voulue ( $V_x, V_y, V_z$ ) en lui faisant subir une rotation autour de  $Oy$  et une autour de  $Oz$  de

$$\begin{aligned} \cos \theta_y &= (V_x^2 + V_y^2)^{1/2} (V_x^2 + V_y^2 + V_z^2)^{-1/2} \\ &\text{avec } -\pi/2 \leq \theta_y \leq 0, \\ \cos \theta_z &= V_x (V_x^2 + V_y^2)^{-1/2} \\ &\text{avec } 0 \leq \theta_z \leq \pi, \end{aligned}$$

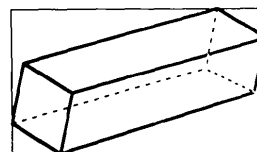


Fig. 3. Rectangle balayé par *LMCTEP* pour le calcul d'un objet contenu dans une boîte.

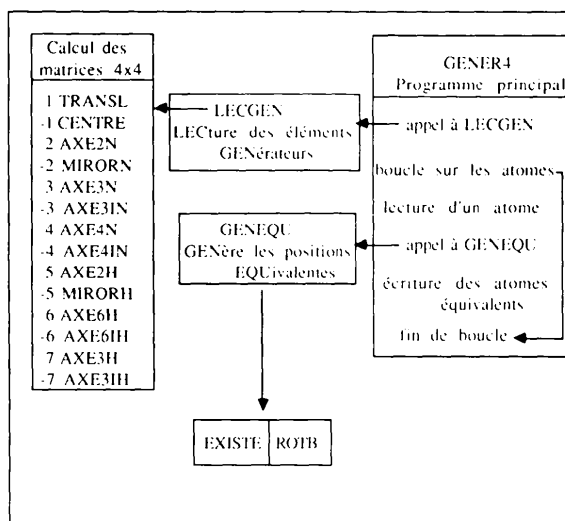


Fig. 2. Organigramme du programme *GENER4*.

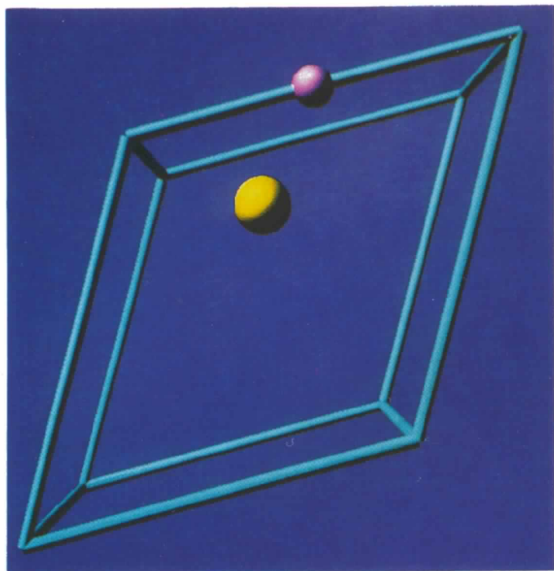


Fig. 4. Unité asymétrique du quartz: silicium et oxygène. Les cylindres représentent les limites de la maille.

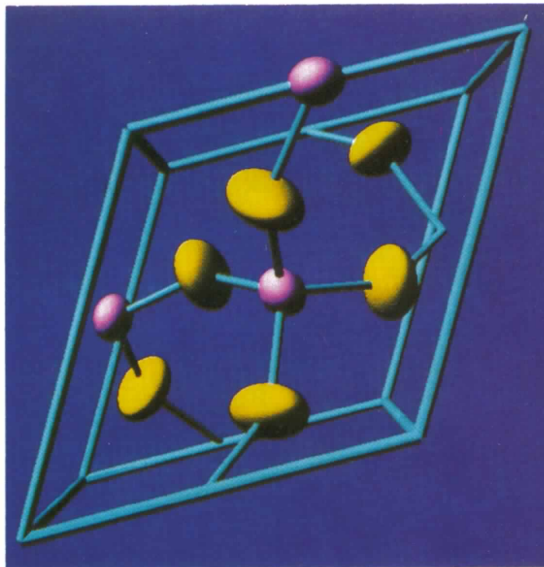


Fig. 5. Contenu d'une maille du quartz avec les liaisons et des ellipsoïdes d'agitation thermique (leur taille a été choisie pour une vue claire et n'a pas ici de signification physique).

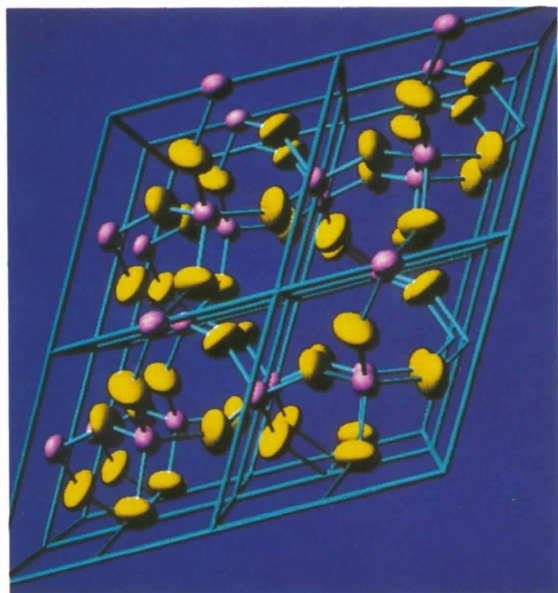


Fig. 6. Contenu de huit mailles du quartz. Temps de calcul dans le Tableau 1.

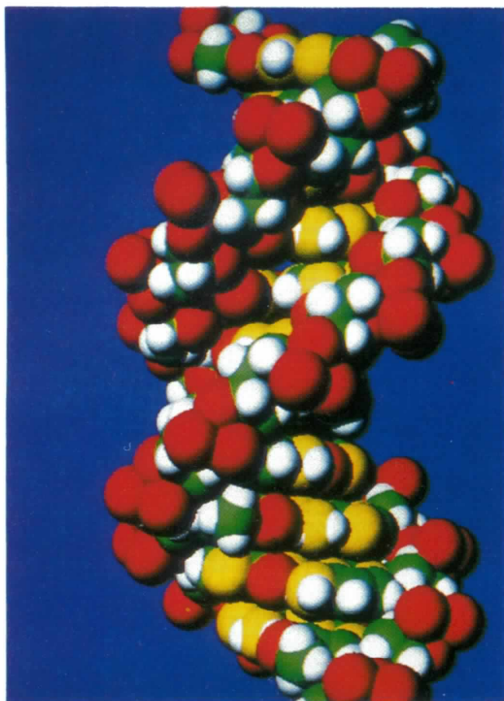


Fig. 7. Vue gauche d'un fragment de l'ADN forme B (729 atomes). Temps de calcul dans le Tableau 1.

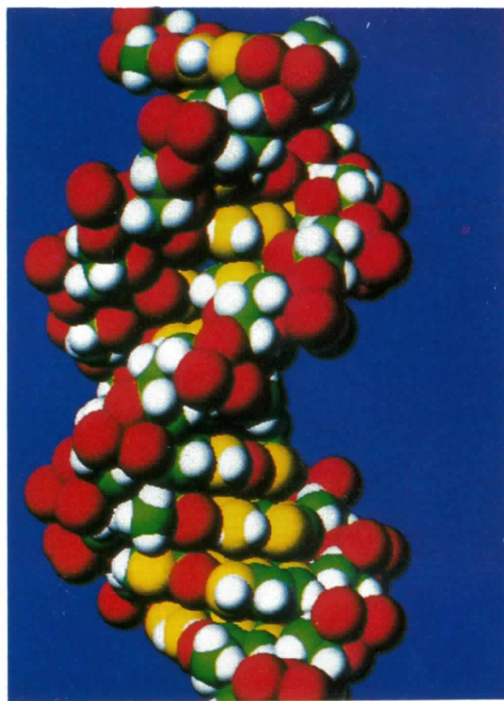


Fig. 8. Vue droite d'un fragment de l'ADN forme B. Forme une paire stéréographique avec la figure précédente.

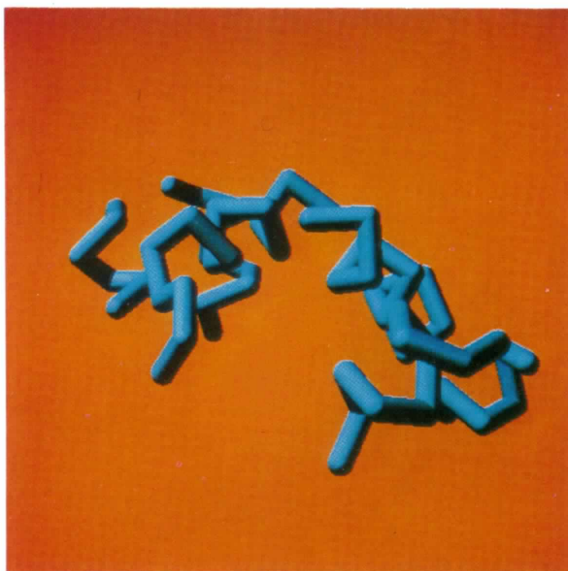


Fig. 9. Hélices de l'insuline représentées en joignant les  $C_\alpha$  par des cylindres.

puis le translate au bon endroit. Ceci est effectué en multipliant la matrice ci-dessus à gauche et à droite par les matrices de transformations adéquates.

De plus *ORTHON* écrit pour chaque objet (sphère, ellipsoïde ou cylindre) les coordonnées des huit sommets d'un parallélépipède entourant cet objet.

La taille mémoire de *ORTHON*, qui comporte 2400 lignes de code source, dépend du nombre maximum d'atomes que l'on permet dans une maille (les liaisons entre atomes, ainsi que les mailles voisines, sont écrites directement sur disque sans être stockées en mémoire). Pour un maximum de 1000 atomes la taille est d'environ 620 kbyte. Le temps d'exécution est évidemment proportionnel au nombre de mailles générées.

### 5. Le programme de représentation *LMCTEP*

Pour chaque objet à représenter le programme lit dans le fichier fabriqué par *ORTHON* une matrice  $4 \times 4$  **Q** et les coordonnées de huit points formant un parallélépipède contenant cet objet. *LMCTEP* va aussi lire un deuxième fichier (Fig. 1) dans lequel l'utilisateur aura précisé des paramètres de la visualisation tels que zone à représenter, distance de l'observateur, angles de rotation, direction de l'éclairage etc.

Les transformations appliquées aux objets sont:

- (a) une translation pour amener le centre de la zone à représenter à l'origine;
- (b) des rotations autour des trois axes  $Ox$ ,  $Oy$ ,  $Oz$ ;
- (c) une mise à l'échelle;
- (d) une transformation perspective;
- (e) une translation pour se ramener dans l'écran.

L'équation, dans le repère orthonormé initial, d'un objet à représenter s'écrit

$$\mathbf{X}^T \mathbf{Q} \mathbf{X} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} & Q_{14} \\ Q_{12} & Q_{22} & Q_{23} & Q_{24} \\ Q_{13} & Q_{23} & Q_{33} & Q_{34} \\ Q_{14} & Q_{24} & Q_{34} & Q_{44} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = 0.$$

Lorsqu'on fait subir à cet objet les transformations précédentes, résumées en une matrice **P**,

$$\mathbf{X}' = \mathbf{P} \mathbf{X},$$

sa nouvelle équation devient

$$\mathbf{X}'^T \mathbf{Q} \mathbf{X} = \mathbf{X}'^T (\mathbf{P}^{-1})^T \mathbf{Q} \mathbf{P}^{-1} \mathbf{X}' = 0,$$

soit  $\mathbf{X}'^T \mathbf{R} \mathbf{X}' = 0$  avec  $\mathbf{R} = (\mathbf{P}^{-1})^T \mathbf{Q} \mathbf{P}^{-1}$ .

*LMCTEP* va balayer tous les pixels contenus dans un rectangle entourant la projection de la boîte contenant l'objet (Fig. 3).

Pour chaque pixel de coordonnées  $x'$ ,  $y'$ , 0 il faut trouver le point de coordonnées  $x'$ ,  $y'$ ,  $z'$  de **R** qui

Tableau 1. Les temps de calcul CPU en secondes pour les Figs. 6 et 7 sur différentes machines

	Norsk Data 570 CX	Sun 4 110	Evans Sutherland ESV20	HP9000 370
Fig. 6				
<i>GENER4</i>	0.4	0.2	0.1	0.3
<i>ORTHON</i>	3.0	1.5	0.9	1.3
<i>LMCTEP</i>	45.6	39.3	8.2	46.7
Fig. 7				
<i>ORTHON</i>	7.9	7.4	3.0	3.5
<i>LMCTEP</i>	111	123	24.2	153

se projette sur ce pixel.  $z'$  est la solution de  $\mathbf{X}'^T \mathbf{R} \mathbf{X}' = 0$ , c'est-à-dire de l'équation du second degré

$$\begin{aligned} &R_{33}z'^2 + 2(R_{13}x' + R_{23}y' + R_{34})z' \\ &+ 2(R_{12}x'y' + R_{14}x' + R_{24}y') \\ &+ (R_{11}x'^2 + R_{22}y'^2 + R_{44}) = 0. \end{aligned}$$

Si le discriminant de cette équation est négatif on est en dehors de l'objet: il n'y a rien à dessiner. Sinon on a deux solutions: on choisit le  $z'$  le plus petit (plus proche de l'observateur). L'élimination des parties cachées se fait par la technique classique d'une mémoire de profondeur (Z-buffer): on ne retient  $z'$  que s'il est inférieur à celui stocké dans le Z-buffer.

Le calcul de l'éclairement se fait de manière simplifiée: on ajoute un premier terme proportionnel au cosinus entre la normale  $\mathbf{R} \mathbf{X}'$  et la lumière incidente (composante diffuse), avec un deuxième terme proportionnel au cosinus entre la direction d'observation et la direction symétrique de l'éclairage par rapport à la normale (composante pseudo-spéculaire).

*LMCTEP*, programme de 900 instructions Fortran, est plus gourmand en mémoire que les programmes précédents à cause des gros tableaux nécessaires pour l'image et le Z-buffer: environ 1800 kbytes.

### 6. Exemples

Toutes les images présentées ici ont une résolution de  $512 \times 512$  et ont été obtenues en photographiant directement l'écran. La Fig. 4 présente un exemple très simple: on y voit un atome de silicium (sur une arête) et un d'oxygène formant l'unité asymétrique du quartz, représentés par des sphères; on a demandé en plus le dessin des limites de la maille à l'aide de cylindres. Le contenu d'une maille a été généré sur la Fig. 5, avec des liaisons et des ellipsoïdes.

La Fig. 6 présente un 'morceau' de quartz comprenant huit mailles: 72 ellipsoïdes et 150 cylindres. A titre indicatif, les temps de calcul CPU en secondes pour cette image et la suivante sont indiqués dans le Tableau 1 sur différentes machines.

Bien que conçu pour visualiser des structures cristallines, le logiciel permet aussi de représenter des molécules isolées comme le fragment d'ADN forme B de la Fig. 7 (729 atomes) qui forme une paire stéréographique avec la Fig. 8. Enfin sur la Fig. 9 on a représenté, à partir d'un fichier de la Protein Data Bank, les hélices de l'insuline en joignant les  $C_\alpha$  par des cylindres.

### 7. Remarques conclusives

Même si notre logiciel n'est pas aussi performant que d'autres déjà existants pour la synthèse d'images formées uniquement de sphères, il possède un certain nombre d'atouts:

- (i) possibilité de représentations réalistes des ellipsoïdes d'agitation thermique;
- (ii) images calculées en perspective vraie (et sans approximation pour les intersections entre sphères, cylindres et ellipsoïdes);
- (iii) ne demande qu'une petite station de travail courante;
- (iv) très bonne portabilité.

Le programme *GENER4* peut être si nécessaire généralisé aux 1651 groupes d'espace bicolores de

Shubnikov & Belov (1964): il faudra pour cela utiliser des matrices  $5 \times 5$ .

Nous tenons à remercier Y. Epelboin pour la mise au point de cet article et A. Jeanne-Michaud pour les photographies.

### Références

- EPELBOIN, Y., GUIDI-MOROSINI, C., MORRIS, F., RIMSKY, A. & SOYER, A. (1986). 2ème Colloque Image du CESTA à Nice, France, Tome 2, pp. 490-496.
- International Tables for X-ray Crystallography* (1983). Tome I, ch. 5, §8.1 et ch. 11. Birmingham: Kynoch Press.
- JOHNSON, C. K. (1965). *ORTEP*. Rapport ORNL-3794. Oak Ridge National Laboratory, Tennessee, E.-U.
- JOHNSON, C. K. (1969). Dans *Crystallographic Computing*, édité par F. R. AHMED, pp. 227-230. Copenhagen: Munksgaard.
- KELLER, E. (1989). *J. Appl. Cryst.* **22**, 19-22.
- LEVY, H. A. (1956). *Acta Cryst.* **9**, 679.
- MAX, N. (1983). *IEEE Comput. Graph. Appl.* **3**, 21-29.
- SHUBNIKOV, A. & BELOV, N. (1964). Dans *Colored Symmetry*, édité par W. HOLSER, pp. 3-210. Oxford: Pergamon Press.
- TRUEBLOOD, K. N. (1956). *Acta Cryst.* **9**, 359-361.